PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-014826

(43)Date of publication of application : 18.01.2002

(51)Int.Cl. G06F 9/45

(21)Application number : 2000-196871

(22)Date of filing : 29.06.2000

(71)Applicant : TOSHIBA CORP

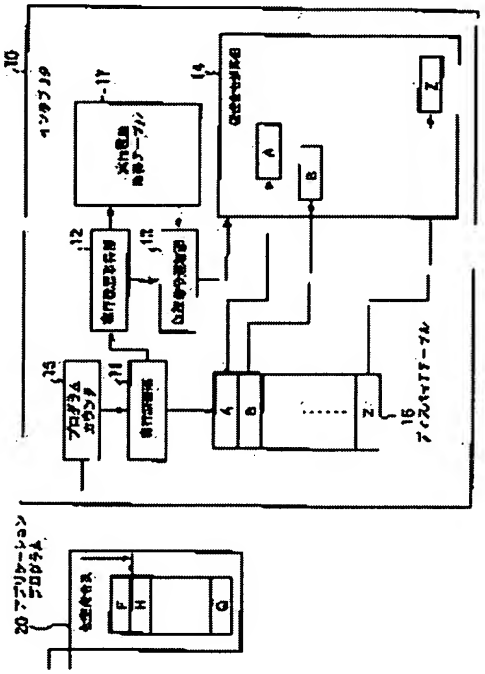
(72)Inventor : SAKAI RYUJI

(54) INTERPRETER AND METHOD FOR EXECUTING ITS PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an interpreter capable of realizing acceleration equivalent to that of JIT compiler with a simpler mechanism without scarifying the starting time of a program.

SOLUTION: An execution control part 11 obtains, from a dispatch table 16, the entry address of a virtual instruction processing part 14 for processing a virtual instruction 21 indicated by a program counter 15 allows the address to jump, and notifies an execution history obtaining part 12 of which virtual instruction is the target for execution control. On the other hand, the execution history obtaining part 12 makes a pair of the virtual instruction and the previously communicated virtual instruction, and monitors the execution frequency by counting the number of times of appearance by using an execution history obtaining table 17, and notifies a virtual instruction connecting part 13 of the connection of the pair whose execution frequency is high as one virtual instruction. Then, the virtual instruction connecting part 13 executes the reconstitution of a virtual instruction series in order to connect the communicated pair as one virtual instruction.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] An activation hysteresis acquisition means to acquire the activation hysteresis of a series of virtual instruction trains which consist of two or more virtual instructions, A detection means to detect said a series of high virtual instruction trains of activation frequency from the activation hysteresis acquired by said activation hysteresis acquisition means, While newly generating the processing section which carries out batch processing of said a series of virtual instruction trains detected by said detection means and assigning the empty virtual instruction of the undefined The interpreter characterized by providing the assigned virtual instruction connection means to reconfigurate said a series of virtual instruction trains so that it may be vacant and interpretation activation may be carried out as a virtual instruction.
[Claim 2] Said virtual instruction connection means is the interpreter according to claim 1 carry out having a means to reconfigurate further said a series of virtual instruction trains so that interpretation activation may be carried out that call appearance should be carried out from the virtual instruction for an address call in the processing section, when the processing section which newly generated and assigned the empty virtual instruction of the undefined fulfills the conditions defined beforehand as the description.
[Claim 3] An activation hysteresis acquisition means to acquire the activation hysteresis of a series of virtual instruction trains which consist of two or more virtual instructions, A detection means to detect while performing said a series of high virtual instruction trains of activation frequency from the activation hysteresis acquired by said activation hysteresis acquisition means, While newly generating the processing section which carries out batch processing of said a series of virtual instruction trains detected by said detection means The interpreter characterized by providing a virtual instruction connection means to reconfigurate said a series of virtual instruction trains so that interpretation activation may be carried out that call appearance of the generated processing section should be carried out from the virtual instruction for an address call.
[Claim 4] The virtual instruction for said address call is an interpreter according to claim 2 or 3 characterized by being what prepared using the empty virtual instruction of the undefined.
[Claim 5] The interpreter according to claim 1, 2, 3, or 4 characterized by having a means to limit the object of optimization.
[Claim 6] The interpreter according to claim 1, 2, 3, or 4 characterized by having according to the object of optimization of the dispatch table which matches a virtual instruction and the processing section, and optimizing a proper to each.
[Claim 7] Acquire the activation hysteresis of a series of virtual instruction trains which consist of two or more virtual instructions, and said a series of high virtual instruction trains of activation frequency are detected from this acquired activation hysteresis. While newly generating the processing section which carries out batch processing of said this detected virtual instruction train of a series of and assigning the empty virtual instruction of the undefined The program execution approach of the interpreter characterized by the thing [reconfigurating said a series of virtual instruction trains so that it may be vacant and interpretation activation may be carried out as a virtual instruction] which assigned.

[Claim 8] The program execution approach of the interpreter according to claim 7 carried out [reconfigurating further said a series of virtual instruction trains so that interpretation activation may be carried out that call appearance of the processing section should be carried out from the virtual instruction for an address call, when the processing section which newly generated and assigned the empty virtual instruction of the undefined fulfills the conditions defined beforehand, and] as the description.

[Claim 9] Acquire the activation hysteresis of a series of virtual instruction trains which consist of two or more virtual instructions, and said a series of high virtual instruction trains of activation frequency are detected from this acquired activation hysteresis. While newly generating the processing section which carries out batch processing of said this detected virtual instruction train of a series of The program execution approach of the interpreter characterized by reconfigurating said a series of virtual instruction trains so that interpretation activation may be carried out that call appearance of the generated processing section should be carried out from the virtual instruction for an address call.

[Translation done.]

* NOTICES *

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention relates to the interpreter simpler than a JIT compiler which plots and comes out and realizes equivalent improvement in the speed, and its program execution approach, without starting the interpreter which carries out interpretation activation of the program described using the virtual instruction, and its program execution approach, and sacrificing the warm-up time of a program especially.
[0002]
[Description of the Prior Art] An interpreter interprets an imagination instruction by software and is a program to perform. An interpreter is mounted by one loop formation which interprets and executes an instruction. A virtual instruction is loaded from memory, the class of instruction is distinguished, and the part of the instruction dispatch processing which branches to processing equivalent to that instruction, and the part which performs an instruction function for every instruction are in this loop formation.
[0003] The program execution by the interpreter becomes about ten times later from several times as compared with the program which compiles equivalent processing to the machine instruction of the microprocessor of a target, and operates. then, the virtual instruction executed in order to make activation by the interpreter into a high speed -- beforehand -- arrangement ****** -- it changes into another instruction or transposing two or more virtual instructions to one virtual instruction which performs equivalent processing is performed. Moreover, the technique of the JIT (Just In Time) compile which performs them after compiling a series of virtual instruction trains in a machine instruction train at the time of program execution also exists.
[0004]
[Problem(s) to be Solved by the Invention] By the way, the approach of performing, while rewriting this virtual instruction to another instruction Since it is necessary to give the regulation and structure for transposing the pair of a specific virtual instruction or a virtual instruction, or the sequence of a virtual instruction to a specific virtual instruction (an empty virtual instruction being used) beforehand to an interpreter Since the rewriting rule was defined only in the range which can define a virtual instruction, there was so big no effectiveness from a viewpoint of improvement in the speed.
[0005] On the other hand, since size becomes large from a virtual instruction train in the machine instruction train and the improvement in the speed by JIT compile needs a lot of memory, it cannot be used by the small inclusion device of memory size in many cases. Moreover, when carrying out JIT compile at the time of activation of the beginning of each batch, since the warm-up time of a program becomes long, use of a JIT compiler is stopped in many cases.
[0006] Moreover, although what is necessary is to carry out JIT compile of not all the processings, but just to carry out JIT compile only of the high processing of activation frequency in a system with small memory size, the problem of using much memory also has the structure which carries out JIT compile in quest of activation frequency.

[0007] It aims at offering the interpreter simpler than a JIT compiler which plots and comes out and realizes equivalent improvement in the speed, and its program execution approach, without making this invention in consideration of such a situation, and sacrificing the warm-up time of a program.
[0008]
[Means for Solving the Problem] In order to attain the purpose mentioned above, the interpreter of this invention The high virtual instruction train of activation frequency is dynamically reconfigurated as one virtual instruction in program execution using the empty virtual instruction generally because of reservation of expandability etc. prepared. To eye others An activation hysteresis acquisition means to acquire the activation hysteresis of a series of virtual instruction trains which consist of two or more virtual instructions, A detection means to detect said a series of high virtual instruction trains of activation frequency from the activation hysteresis acquired by said activation hysteresis acquisition means, While newly generating the processing section which carries out batch processing of said a series of virtual instruction trains detected by said detection means and assigning the empty virtual instruction of the undefined It is characterized by providing the assigned virtual instruction connection means to reconfigurate said a series of virtual instruction trains so that it may be vacant and interpretation activation may be carried out as a virtual instruction.
[0009] In the interpreter of this invention, the thing simpler than a JIT compiler for which it plots and comes out and equivalent improvement in the speed is realized is made possible, without sacrificing the warm-up time of a program for n times of the loop formations which are needed for that activation about the high virtual instruction train of activation frequency by reconfigurating dynamically so that it may end with 1 time of a loop formation.
[0010] Moreover, the interpreter of this invention carries out having a means reconfigurate further said virtual instruction trains of a series of so that interpretation activation may be carried out that call appearance should be carried out from the virtual instruction for an address call in that processing section as the description, when the processing section which said virtual instruction connection means newly generated, and assigned the empty virtual instruction of the undefined fulfills the conditions defined beforehand.
[0011] It is ** about using a limited empty virtual instruction effectively in the interpreter of this invention for it to be possible.
[0012] Moreover, the interpreter of this invention is characterized by the virtual instruction for said address call being what is prepared using the empty virtual instruction of the undefined.
[0013] In the interpreter of this invention, even if it is the case where the virtual instruction for an address call is not defined beforehand, it makes it possible to perform optimization by this technique.
[0014] Moreover, the interpreter of this invention is characterized by having a means to limit the object of optimization.
[0015] In the interpreter of this invention, for example, it is made to correspond to a program execution plan etc., and makes it possible to make effectiveness of optimization easy to acquire.
[0016] Moreover, the interpreter of this invention is characterized by having according to the object of optimization of the dispatch table which matches a virtual instruction and the processing section, and optimizing a proper to each.
[0017] the interpreter of this invention -- setting -- a method, a function, or a batch -- ** -- it makes it possible to optimize by being alike.
[0018]
[Embodiment of the Invention] Hereafter, 1 operation gestalt of this invention is explained with reference to a drawing.
[0019] Drawing 1 is a conceptual diagram for explaining the program execution approach of the interpreter concerning the operation gestalt of this invention, and that optimization also carries out dynamically in the interpreter 10 by each processing section of the execution control section 11, the activation hysteresis acquisition section 12, the virtual instruction connection section 13, and the virtual instruction-processing section 14, and each data division of a program counter 15, a dispatch table 16, and the activation hysteresis acquisition table 17, carrying out

interpretation activation in the virtual instruction train 21 of an application program 20.
[0020] In addition, rewriting of the virtual instruction train 21 which the interpreter 10 and application program 20 which are shown in this drawing 1 are loaded to the main memory of a computer from external memory, such as a hard disk, for example, is mentioned later etc. is performed only on main memory to the last.
[0021] The execution control section 11 loads the virtual instruction 21 to which a program counter 15 points, acquires the entry address of the virtual instruction-processing section 14 for processing the virtual instruction 21 from a dispatch table 16, and moves control to the entry address of the acquired virtual instruction-processing section 14 (it jumps).
[0022] Although this dispatch table 16 is for matching each virtual instruction and the entry address of the virtual instruction-processing section 14, as shown in drawing 2 , it is equipped with the free area for storing the entry address of a new virtual instruction for securing expandability etc. in addition to the field which stores the entry address of the existing virtual instruction defined beforehand. Moreover, in this interpreter 10, one of the virtual instructions newly established by that free area is reserved as a native procedure instruction (processing which jumps the entry address to reception and its entry address as an operand is performed) mentioned later.
[0023] Moreover, it notifies about which virtual instruction the execution control section 11 carried out execution control to the activation hysteresis acquisition section 12 at this time.
[0024] On the other hand, the activation hysteresis acquisition section 12 which received this notice makes a pair the virtual instruction notified before this notified virtual instruction and one, and counts that count of an appearance using the activation hysteresis acquisition table 17. If drawing 3 is drawing showing an example of the activation hysteresis acquisition table 17, for example, the virtual instruction with which the virtual instruction notified this time was notified before one of them by H is F, the activation hysteresis acquisition section 12 By incrementing the value of a F line H train (F, H), conversely, if the virtual instruction with which the virtual instruction notified this time was notified before one of them by F is H, the activation hysteresis acquisition section 12 will count the count of an appearance of each pair by incrementing the value of a H line F train (H, F).
[0025] The activation hysteresis acquisition section 12 has the values s1 and s2 for having threshold S and the sample period T, and controlling this threshold S (s1<s2). And the activation hysteresis acquisition section 12 will notify that to the virtual instruction connection section 13 that these should be connected as one virtual instruction, if threshold S is initialized by s1 and the count of an appearance of one of pairs exceeds threshold S first. In addition, the activation hysteresis acquisition section 12 reinitializes threshold S by s1 while it will initialize all the counted value of the activation hysteresis acquisition table 17 by 0 if it adds s1 to threshold S and serves as threshold S>s2 whenever the sample period T passes. This realizes connecting a virtual instruction more accommodative.
[0026] And the virtual instruction connection section 13 performs reconstruction of a virtual instruction train that the pair of the virtual instruction notified from the activation hysteresis acquisition section 12 should be connected with one virtual instruction. Hereafter, an example of this virtual instruction connection is explained with reference to drawing 4 thru/or drawing 7 .
[0027] Now, the count of an appearance of the pair (G, R) of the virtual instruction G and the virtual instruction R exceeds threshold S, and it is assumed that these connection was notified from the activation hysteresis acquisition section 12 to the virtual instruction connection section 13. Drawing 4 is drawing showing the condition before connection of a pair (G, R) is performed.
[0028] As shown in drawing 4 , in the condition before connection, the virtual instruction train 21 of an application program 20 is dotted with many pairs (G, R), and, on the other hand, the entry address of the processing section G of the normal of the virtual instruction-processing section 14 and the processing section R is stored in the dispatch table 16 of an interpreter 10 for the virtual instruction G and the virtual instruction R.
[0029] In case a pair (G, R) is connected, as shown in drawing 5 , the virtual instruction connection section 13 First, the processing section C in which the degree of the virtual instruction G confirms whether to be the virtual instruction R (G, R) The processing section G

(G, R) which generates the processing section J (G, R) which connected the virtual instruction G and the virtual instruction R, and the processing section W for instruction rewriting (G, R) is generated in the virtual instruction-processing section 14. The entry address of the virtual instruction G stored in a dispatch table 16 is rewritten from the entry address of the processing section G to the entry address of the processing section C (G, R). Drawing 5 shows the condition in the time of completing this processing.

[0030] Next, when a program counter 12 points to the virtual instruction G of a pair (G, R) in this condition, a dispatch table 16 will be referred to by the execution control section 11, and the jump to the rewritten processing section C (G, R) will be performed.

[0031] The fetch of the next virtual instruction is carried out from the execution control section 11, and if the virtual instruction is R, if the processing section C (G, R) is not so in the processing section G (G, R), in it It jumps in the processing section G, and since a degree is the virtual instruction R here (the copy of the processing section G may be combined instead of jumping in the processing section G), it jumps in the processing section G (G, R).

[0032] As shown in drawing 6 , on the other hand, the processing section G (G, R) First, the processing section J (G, R) which connected the processing section G and the processing section R is generated in the virtual instruction-processing section 14. While acquiring and assigning one virtual instruction from the pool of an empty virtual instruction of a dispatch table 16 (this is set to J (G, R)), it adds to the matrix of the activation hysteresis acquisition table 17 that this pair should be made applicable [of the activation hysteresis acquisition section 12] to a count.

[0033] Next, the processing section G (G, R) generates the processing section W (G, R) for rewriting the virtual instruction pair (G, R) of the application program [INTAPURITO / application program] 20 in the virtual instruction-processing section 14, and rewrites branching to the processing section G (G, R) from the processing section C (G, R) to branching to the processing section W (G, R).

[0034] Finally, the processing section G (G, R) is jumped to the processing section W (G, R) (since the processing section W (G, R) is finite processing realizable if the processing section G and the processing section R are given as a numeric parameter, you may prepare before activation). Drawing 6 shows the condition in the time of completing this processing.

[0035] Moreover, as shown in drawing 7 , in the processing section W of a jump place (G, R), the virtual instruction G to which the program counter 12 of KARENTO of the virtual instruction train [INTAPURITO / train] 21 is pointing is rewritten to J (G, R), and the next virtual instruction R is rewritten to an operand, and it jumps in the processing section J (G, R). And it is collectively performed by the virtual instruction pair (G, R) in the processing section J (G, R). Drawing 7 shows the condition in the time of completing this processing.

[0036] Henceforth, whenever the virtual instruction pair (G, R) of the virtual instruction train 21 appears, the jump to the processing section W (G, R) will be performed from the processing section C (G, R), rewriting of the virtual instruction train 21 and package activation of a virtual instruction pair (G, R) will be repeated, and all virtual instruction pairs (G, R) will be someday rewritten by J (G, R). Moreover, when virtual instructions other than the virtual instruction R come to the degree of the virtual instruction G, from the processing section C (G, R), it jumps in the processing section G and is processed satisfactory at all. Furthermore, if it is after the completion of rewriting, it is possible to perform same processing, transposing the entry address of the virtual instruction G with which the virtual instruction G overlaps and which is stored in a dispatch table 16 also about a virtual instruction pair (G, S) from the entry address of the processing section J (G, R) to the entry address of the processing section J (G, S), for example.

[0037] Thus, the thing simpler than a JIT compiler for which it plots and comes out and equivalent improvement in the speed is realized is made possible, without according to the program execution approach of this interpreter, sacrificing the warm-up time of a program for n times of the loop formations which are needed for that activation by reconfigurating dynamically about the high virtual instruction train of activation frequency, so that it may end with 1 time of a loop formation.

[0038] Moreover, the count of an appearance of a pair with other virtual instructions will count

the virtual instruction J (G, R) generated by this connection, and connection to the virtual instruction of further others will be advanced depending on the activation situation of an application program 20.

[0039] By the way, if connection is advanced in this way, the virtual instruction train of a part with high activation frequency will become one virtual instruction someday. Then, when the number of the original virtual instructions which it is going to connect with one virtual instruction exceeds a predetermined threshold or serves as the basic block whole region, the virtual instruction connection section 13 applies optimization to the connected batch so that the performance may become a high speed more. A basic block here means each part which is made when a program is divided at the branching place of branch instruction and branch instruction and which is surely performed sequentially.

[0040] If the processing section of this virtual instruction has become throughout the basic block at this time, the virtual instruction connection section 13 investigates an adjoining basic block, an adjoining basic block is also one virtual instruction, and when the number of an inlet port and outlets is one, respectively, it will generate the processing section which optimized processing of the basic block of these plurality about processing of this whole as one virtual instruction about the continuation field of these basic blocks. Since the optimized processing section does not need the information on operand part any longer, when it becomes more than the die length which needs the die length of the operand part of the connected virtual instruction for the address expression of the procedure of the microprocessor of a target, it rewrites to the native procedure-call instruction (processing which jumps the entry address to reception and its entry address as an operand is performed) which mentioned this virtual instruction above. Hereafter, an example of this virtual instruction rewriting is explained with reference to drawing 8 and drawing 9 .

[0041] The virtual instructions J1-J4 newly generated by connection shall exist now, and the virtual instruction J1, the virtual instruction J2, and the virtual instruction J3 and the virtual instruction J4 shall be connected further. And both sides shall fulfill the conditions for rewriting to a native procedure-call instruction. Drawing 8 is drawing showing the condition before this rewriting.

[0042] it is shown in drawing 8 -- as -- the condition before rewriting -- the virtual instruction train 21 of an application program 20 -- a pair (J1, J2) and a pair (J3, J4) -- being dotted -- **** -- on the other hand -- the dispatch table 16 of an interpreter 10 -- the virtual instructions J1-J4 -- it is alike, respectively and the entry address of the new processing sections J1-J4 of the virtual instruction-processing section 14 is stored.

[0043] As shown in drawing 9 , and the virtual instruction connection section 13 The processing section J (J3+J4) which connected the virtual instructions J3-J4 with the processing section J (J1+J2) which connected the virtual instructions J1-J2 is generated in the virtual instruction-processing section 14. First, subsequently It rewrites to the native procedure-call instruction which had the entry address of the processing section J (J1+J2) by making into an operand the pair (J1, J2) with which the virtual instruction train 21 is dotted. Moreover, it rewrites to the native procedure-call instruction which had the entry address of the processing section J (J3+J4) by making into an operand the pair (J3, J4) with which the virtual instruction train 21 is dotted. Drawing 9 shows the condition in the time of completing this processing.

[0044] When further improvement in the speed of the connected batch is attained by this and the activation frequency of J1-J4 falls, it becomes possible to use an empty virtual instruction limited as a result effectively.

[0045] Next, with reference to drawing 10 and drawing 11 , the operations sequence of the program execution approach of this interpreter 10 is explained.

[0046] This interpreter 10 loads the virtual instruction 21 to which a program counter 15 points [the execution control section 11] (step A1 of drawing 10), and the activation hysteresis acquisition section 12 acquires the activation hysteresis of a virtual instruction train to coincidence (step A2 of drawing 10).

[0047] Moreover, the activation hysteresis acquisition section 12 supervises whether the virtual instruction train in which the count of an appearance exceeds a threshold exists (step A3 of

drawing 10), and if existence of the virtual instruction train exceeding a threshold is detected (YES of step A3 of drawing 10), it will make the virtual instruction connection section 13 perform connection of the virtual instruction train (step A4 of drawing 10).
[0048] On the other hand, the virtual instruction connection section 13 [whether the number of virtual instructions of the basis of the virtual instruction train which should be connected exceeds a threshold, and] Or it judges whether it is a basic block (step B1 of drawing 11). When the number of virtual instructions of a basis exceeds a threshold or it has become a basic block (YES of step B1 of drawing 11), reconstruction of a virtual instruction train is performed that this virtual instruction train that should be connected should be considered as a native procedure-call instruction (step B-2 of drawing 11).
[0049] Moreover, if the virtual instruction connection section 13 tries acquisition of an empty virtual instruction (step B3 of drawing 11) and it cannot acquire probably when the number of virtual instructions of a basis does not exceed a threshold and does not serve as a basic block, either (NO of step B1 of drawing 11) (NO of step B4 of drawing 11), the lowest empty virtual instruction of activation frequency is released (step B5). This release is carried out by returning the virtual instruction train 21 and dispatch table 16 which were rewritten for that empty virtual instruction. Then, the virtual instruction connection section 13 performs reconstruction of a virtual instruction train that the virtual instruction train which should be connected should be considered as the acquired empty virtual instruction (step B6 of drawing 11).
[0050] And after jumping the execution control section 11 to the entry address of the virtual instruction-processing section 14 acquired from the dispatch table 16 (step A5 of drawing 10) and incrementing a program counter 12 by the size of the virtual instruction (drawing 10 step A6), it repeats the above-mentioned processing.
[0051] Thus, the thing simpler than a JIT compiler for which it plots and comes out and equivalent improvement in the speed is realized is made possible, without according to the program execution approach of this interpreter, sacrificing the warm-up time of a program for n times of the loop formations which are needed for that activation by reconfigurating dynamically about the high virtual instruction train of activation frequency, so that it may end with 1 time of a loop formation.
[0052] In addition, although the situation explained the example using a native procedure call instruction with the above-mentioned operation gestalt on the principle of performing connection of a virtual instruction train using an empty virtual instruction first, you may perform according to a resource environment etc. using a native procedure call instruction from the beginning.
[0053] Moreover, what is necessary is just to use the existing virtual instruction, when the virtual instruction for an address call is defined beforehand although the above-mentioned operation gestalt explained the example which is vacant and prepares a native procedure call instruction using a virtual instruction.
[0054] By the way, there are few virtual instructions, and when an empty virtual instruction exists enough and an empty virtual instruction does not exist enough although it is possible for processing of a part like a small loop formation collected to some extent to become one virtual instruction, or to become a native procedure-call instruction, and to obtain the engine performance equivalent to a JIT compiler, a possibility that sufficient effectiveness may not be acquired is. Then, if it has further the function which limits the object which extracts and optimizes activation hysteresis to procedure to accelerate then, for example, it will be made to correspond to a program execution plan etc., and it will become possible to make effectiveness of optimization easy to acquire.
[0055] furthermore -- if a method, a function, or a batch uses a different dispatch table 16 for every object of optimization -- this method, a function, or a batch -- ** -- it becomes possible to be alike and to perform virtual instruction connection of a proper.
[0056]
[Effect of the Invention] As explained in full detail above, according to the interpreter of this invention, the empty virtual instruction generally because of reservation of expandability etc. prepared is used. The high virtual instruction train of activation frequency from having made it reconfigurate as one virtual instruction dynamically in program execution The thing simpler than

a JIT compiler for which it plots and comes out and equivalent improvement in the speed is realized is made possible, without sacrificing the warm-up time of a program for n times of the loop formations which are needed for the activation by reconfigurating dynamically about the high virtual instruction train of activation frequency, so that it may end with 1 time of a loop formation.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] The conceptual diagram for explaining the program execution approach of the interpreter concerning the operation gestalt of this invention.
[Drawing 2] Drawing showing the configuration of the dispatch table with which the interpreter of this operation gestalt is equipped.
[Drawing 3] Drawing showing the configuration of the activation hysteresis acquisition table with which the interpreter of this operation gestalt is equipped.
[Drawing 4] Drawing 1 for explaining an example of the virtual instruction connection which the interpreter of this operation gestalt performs.
[Drawing 5] Drawing 2 for explaining an example of the virtual instruction connection which the interpreter of this operation gestalt performs.
[Drawing 6] Drawing 3 for explaining an example of the virtual instruction connection which the interpreter of this operation gestalt performs.
[Drawing 7] Drawing 4 for explaining an example of the virtual instruction connection which the interpreter of this operation gestalt performs.
[Drawing 8] Drawing 1 for explaining an example of rewriting to the native procedure-call instruction which the interpreter of this operation gestalt executes.
[Drawing 9] Drawing 2 for explaining an example of rewriting to the native procedure-call instruction which the interpreter of this operation gestalt executes.
[Drawing 10] The 1st flow chart for explaining the operations sequence of the program execution approach of the interpreter of this operation gestalt.
[Drawing 11] The 2nd flow chart for explaining the operations sequence of the program execution approach of the interpreter of this operation gestalt.
[Description of Notations]
10 -- Interpreter
11 -- Execution control section
12 -- Activation hysteresis acquisition section
13 -- Virtual instruction connection section
14 -- Virtual instruction-processing section
15 -- Program counter
16 -- Dispatch table
17 -- Activation hysteresis acquisition table
20 -- Application program
21 -- Virtual instruction train

[Translation done.]